Data Mining for Multi-agent Fuzzy Decision Tree Structure and Rules

Dr. James F. Smith III
Naval Research Laboratory, Code 5741
Washington, D.C., 20375-5000
Telephone: 202.767.5358
Fax: 202.404.7690
jfsmith@drsews.nrl.navy.mil

Abstract- *A fuzzy logic based expert system has been developed that automatically allocates resources in real-time over many dissimilar platforms. The platforms can be very general, e.g., ships, planes, etc. Potential foes can also be general. The resource manager has been embedded in an electronic game environment. This co-evolutionary game fully automates the data mining problem allowing determination of parameters essential to the resource manager. The game allows the resource manager to learn from human experts or computerized enemies. The game does not determine the structure of fuzzy decision trees. A new data mining algorithm that uses a genetic program, an algorithm that evolves other computer programs, as a data mining function has been developed to solve this problem. It not only determines the fuzzy decision tree structure it also creates fuzzy rules while mining scenario data bases. Finally, experimental results are discussed related to both data mining algorithms.*

Keywords: data mining, knowledge discovery, fuzzy logic, genetic algorithms, genetic programs, resource manager, expert systems

1 Introduction

Modern naval battleforces generally include many different platforms, e.g., ships, planes, helicopters, etc. Each platform has its own sensors, e.g., radar, electronic support measures (ESM), and communications. The sharing of information measured by local sensors via communication links across the battlegroup should allow for optimal or near optimal decisions. The survival of the battlegroup or members of the group depends on the automatic real-time allocation of various resources.

A fuzzy logic algorithm has been developed that automatically allocates electronic attack (EA) resources in real-time. In this paper EA refers to the active use of electronic techniques to neutralize enemy equipment such as radar [1]. The particular approach to fuzzy logic that will be used is the fuzzy decision tree, a generalization of the standard artificial intelligence technique of decision trees [2].

The controller must be able to make decisions based on rules provided by experts. The fuzzy logic approach allows the direct codification of expertise forming a fuzzy linguistic description [3], i.e., a formal representation of the system in terms of fuzzy if-then rules. This will prove to be a flexible structure that can be extended or otherwise altered as doctrine sets, i.e., the expert rule sets change.

The fuzzy linguistic description will build composite concepts from simple logical building blocks known as root concepts through various logical connectives: "or", "and", etc. Optimization has been conducted to determine the form of the membership functions for the fuzzy root concepts and fuzzy decision tree structure.

The optimization procedures employed here are a type of data mining. Data mining is defined as the efficient discovery of valuable, non-obvious information embedded in a large collection of data [4]. The genetic optimization techniques used here are efficient, the relationship between quantities extracted and the fuzzy rules are certainly not a priori obvious, and the information obtained is valuable for decision-theoretic processes. Also, the genetic algorithm based procedure is designed so that when the scenario databases change as a function of time, then the algorithm can automatically re-optimize allowing it to discover new relationships in the data. Alternatively, the resource manager (RM) can be embedded in a computer game that EA experts can play. The software records the result of the RM and expert's interaction, automatically assembling a database of scenarios. After the end of the game, the RM makes a determination of whether or not to re-optimize itself using the newly extended database.

To be consistent with terminology used in artificial intelligence and complexity theory [5], the term

"agent" will sometimes be used to mean platform, also a group of allied platforms will be referred to as a "meta-agent." Finally, the terms "blue" and "red" will refer to "agents" or "meta-agents" on opposite sides of a conflict, i.e., the blue side and the red side.

Section 2 briefly introduces the ideas of fuzzy set theory, fuzzy logic, fuzzy decision trees, and the five major components of the RM. Section 3 discusses optimization with a focus on genetic algorithms and data mining. Section 4 discusses co-evolutionary optimization and its effectiveness. Section 5 provides references to examples of the RM's responses for multi-platform scenarios that provide validation of the RM and the information data mined. Section 6 introduces a different data mining algorithm that uses a genetic program to data mine a military data base. Section 7 describes the application of this algorithm to data mining fuzzy decision tree structure, i.e., how vertices and edges are connected and labeled in a fuzzy decision tree. This is equivalent to automatically generating fuzzy if-then rules. Section 7 concludes with a discussion of experimental results. Finally, section 8 provides a summary.

2 A brief introduction to fuzzy sets, fuzzy logic and the fuzzy RM

The RM must be able to deal with linguistically imprecise information provided by an expert. Also, the RM must control a number of assets and be flexible enough to rapidly adapt to change. The above requirements suggest an approach based on fuzzy logic. Fuzzy logic is a mathematical formalism that attempts to imitate the way humans make decisions. Through the concept of the grade of membership, fuzzy set theory and fuzzy logic allow a simple mathematical expression of uncertainty [6]. The RM requires a mathematical representation of domain expertise. The decision tree of classical artificial intelligence provides a graphical representation of expertise that is easily adapted by adding or pruning limbs. The fuzzy decision tree, a fuzzy logic extension of this concept, allows easy incorporation of uncertainty as well as a graphical codification of expertise [2]. Finally, a detailed discussion of the particular approach to fuzzy logic and fuzzy decision trees used in the RM is given in the literature [7].

The resource manager is made up of five parts, the isolated platform model, the multi-platform model, the communication model, the fuzzy parameter selection tree and the fuzzy strategy tree. As previously discussed the isolated platform model provides a fuzzy decision tree that allows an individual platform to respond to a threat. The multi-platform model allows a group of platforms to respond to a threat in a collaborative fashion. The communication model describes the means of communication or interaction between the platforms. The fuzzy parameter selection tree is designed to make optimal or near optimal selections of root concept parameters from the parameter database assembled during previous optimization with the genetic algorithm. Finally, the strategy tree is a fuzzy tree that an agent uses to try to predict the behavior of an enemy. A more detailed discussion of the structure of the RM as well as explicit forms for fuzzy membership functions can be found in the literature [7].

3 Optimization of the root concept's parameters using a genetic algorithm for data mining

The parameters of the root concept membership function are obtained by optimizing the RM over a database of scenarios using a genetic algorithm [8] (GA). Once the root concept membership functions are known, those for the composite concepts [7] follow immediately. At this point the necessary fuzzy if-then rules for the RM have been fully determined. Detailed discussions of the GA used for data mining as well as the construction of the chromosomes and fitness functions are given in the literature [7].

The optimization procedures employed here are a component of a data mining operation. Data mining is defined as the efficient discovery of valuable, non-obvious information embedded in a large collection of data [4]. The genetic optimization techniques used here are efficient, the relationship between parameters extracted and the fuzzy rules are certainly not a priori obvious, and the information obtained is valuable for decision-theoretic processes. Also, the RM is designed so that when the scenario databases change as a function of time then the algorithm can automatically re-optimize allowing it to discover new relationships in the data.

The application of the genetic algorithm is actually part of the second step in a three-step data mining process. The first step is the collection of data and its subsequent filtering by a domain expert, to produce a scenario database of good quality. The second step involves the use of various data mining functions such as clustering and association, etc. During this step, the genetic algorithm based optimization is used to mine parameters from the database. These parameters allow the fuzzy decision tree to form optimal conclusions about resource allocation. In the third and final step of the data mining operation, the RM's decisions are analyzed by a domain expert to determine their validity.

Data mining to re-optimize the RM and real-time operation of the RM may occur simultaneously on different computers. Since the RM is designed to operate on a collection of platforms, even during very active use of the RM, some computer resources may be available for additional optimization and other data mining related

activities. Thus, the multi-platform scheme allows frequent re-optimization of the RM, while the previously optimized version of the RM continues to function in real-time.

Typically the database is constructed from data taken from sensors of different types. The data will be sparse intermittent and noisy. To assemble a representative database, the domain expert must eliminate unacceptable data followed by the use of various data mining functions such as clustering [9-12], association [13-20], etc. Clustering can be used to organize the data, suppress outliers, etc. Association determines when data measured on different sensors corresponds to the same observable.

An alternate approach to constructing a database for re-optimization involves embedding the RM in a computer game. The game is designed so human EA experts can play it, in real-time against the RM. The game also allows the RM to be matched against computerized opponents running under their own autonomous logic. The game software records the events of the game for both cases, i.e., when the RM's opponent is a human expert or a computerized agent. This record contributes to a database for re-optimization. Such a database is purer than one born of sensor data since environmental noise, sensor defects, etc., are not contaminating the data. This offers the advantage that the filtering stage of the data mining operation is simplified. The obvious disadvantage is that the database will be less representative of events in the real world, than one born of real sensor data taken during battle.

4 Co-evolutionary data mining

In nature a system never evolves separately from the environment that contains it. Both biological system and environment simultaneously evolve. This is referred to as co-evolution [21]. In a similar manner the fuzzy resource manager should not evolve separately from its environment, i.e., enemy tactics should be allowed to simultaneously evolve. Certainly, in real world situations if the enemy sees the resource manager employ a certain range of techniques, they will evolve a collection of counter techniques to compete more effectively with the resource manager.

4.1 Real-time co-evolutionary data mining

The approach to co-evolution is as follows. For each root concept membership function on the red strategy tree define a threshold, such that if the membership function exceeds this threshold and if red's strategy tree is a good representation of blue's decision tree, then red's intention is signaled to blue resulting in an action by blue. The membership function is typically a function of some physically measurable quantity O and its first derivative in time, dO/dt. The two dimensional space resulting from plotting dO/dt vs. O is a phase space. The inequality between the root concept membership function and its threshold, upon inversion will give inequalities linear in O and dO/dt, typically. The resulting system of inequalities defines a region of phase space referred to as the admissible region where red can engage in activities without signaling its intent to blue. The membership function parameters that are found through data mining determine the boundaries of the admissible region of phase space. The admissible region can not in general be brought to zero area otherwise blue will carry out an action against everything it detects, resulting in fratricide and wasting valuable resources essential to its survival.

4.2 Tools for visualization of data mined information

To facilitate data mining, co-evolution and validation of the RM, a software tool known as the scenario generator (SG) has been created. It automatically creates simulated blue and red platforms with user defined assets. It also creates a map or battlespace and automatically places the red and blue platforms in this space where they can interact. Each blue platform is controlled by its own copy of the fuzzy RM.

The SG has two modes of operation. In the computer vs. computer (CVC) mode each red platform is controlled by its own controller distinct from the fuzzy RM used by the blue platforms. In the second mode, the human vs. computer (HVC) mode, a human player controls a red platform through an interactive graphical user interface (GUI). There can be multiple red platforms. At each time step, the human player can control any of the red platforms, but only one of them per time step. Those red platforms not under human control run under their own logic as in the CVC mode.

From the SG software three different GUI's can be easily accessed. These GUI's are the "scenario builder", the "map builder," and the "human control player interface" (HCPI).

The scenario builder GUI allows the construction of blue and red agents with general characteristics. Through this GUI both blue and red agents can be given various assets such as different types of radars, ESM, EA systems, etc.

4.3 Effectiveness of co-evolutionary optimization

One simple class of experiments that has been conducted consists of one blue agent versus one red agent. It was typically found that all three parameters in blue's version of "close" showed little change for the last 33% of

the co-evolutionary generations. The human opponent operating the red agent tended to fixate on the same strategies. This suggested that in HVC optimization the human player quickly reached the limits of his or her expertise resulting in the RM's parameters reaching a constant value. Thus the optimization of blue converged rapidly.

In a simple experiment with one blue agent versus one red agent operating in CVC mode convergence was not nearly as fast as in HVC mode. The computer controlled red agent is typically capable of exhibiting many more strategies than the human controlled red agent in HVC mode. Thus the co-evolutionary process ends up exploring the combined red-blue parameter space longer, resulting in a greater likelihood of a global maximum being found for the fitness function, resulting in a RM that is more robust than in the HVC case.

The more robust RM obtained through use of the CVC optimization can be understood intuitively as follows. If red can exhibit more strategies by using CVC mode than in HVC mode then the blue RM is forced to be more adaptive to compete.

There is a risk during co-evolution that with both red and blue co-evolving, they will become very specialized in dealing with each other. For example without taking proper precautions blue agents of the 1000^{th} co-evolutionary generation might be effective against red agents of that generation, but ineffective against agents of generations 100 through 999. Fortunately, the structure of the symbolically recursive fitness function prevents this, because its form retains the past history of the agents, forcing the blue agents of the 1000^{th} generation to be effective against red agents of the preceding or current generation.

5 Validation of the RM and information data mined

The third step of the data mining problem involves validation, i.e., determination of the value of the information data mined. This is intrinsically coupled to the validation of the resource manager itself. Both the data mined information and the RM have been subjected to significant evaluations using the scenario generator [22-28]. Through this process the data mined information has been shown to be extremely valuable and the decisions made by the RM of the highest quality. Currently, additional validation efforts are underway that involve the use of a hardware simulator with real sensors as opposed to the digitally simulated sensors created by the scenario generator. The results of this hardware validation effort will be the subject of a future publication.

6 Data mining for fuzzy decision tree structure using a genetic program

In section 4 a GA was used as a data mining function to determine parameters for fuzzy membership functions. This section introduces a different data mining function, a genetic program [29] (GP). The GP data mines fuzzy decision tree structure, i.e., how vertices and edges are connected and labeled in a fuzzy decision tree. Whereas the GA based data mining procedures determine the parameters of and hence the form of fuzzy membership functions, the GP based procedure actually data mines fuzzy if-then rules.

6.1 Structure of the genetic program

A genetic program is a problem independent method for automatically creating computer programs. Like a genetic algorithm, it evolves a solution using Darwin's principle of survival of the fittest. Unlike the genetic algorithm, of which it can be considered an extension: its initial, intermediate, and final populations are computer programs.

Like a genetic algorithm the fittest individuals in the population are copied and subject to two operations, crossover and mutation. Crossover corresponds to sexual recombination, a kind of mating between parent computer programs. The crossover operation is constrained to produce structurally valid offspring. Finally, the mutation operation is a random change in a computer program that is part of the evolving population.

A genetic program requires the completion of five major steps before it can be used. The first step involves specifying a set of terminals. The terminals are the actual variables of the problem. These can include a variable like "x" used as a symbol in building a polynomial and also real constants. In the case the computer programs to be built are actually fuzzy decision trees, the terminal set might consist of root concepts that label the leaves of the tree. This is discussed in more detail in section 7.

The second step involves the specification of a set of functions. The functions can consist of operations like addition, multiplication, etc., in the case the computer programs to be assembled are polynomials. If the computer programs to be created are fuzzy decision trees then the functions might consists of logical operations like "and" and "or" and logical modifiers like "not".

The third step consists of specifying the fitness function. The fitness function for a genetic program has essentially the same role as that for a genetic algorithm and may be assembled through a similar process of intuition or derivation.

The fourth step consists of specifying control parameters. These can include the probabilities of

mutation and crossover as well as parameters that relate to the fifth major step.

The fifth and final step consists of determining the termination criteria. This is frequently a maximum number of generations or that the fitness has not changed by a certain amount in a certain number of generations.

Figure 1 provides a flow chart of a typical genetic program. It starts with an initial population that is randomly generated or user supplied. Once the initial population is created, the fitness of each individual is determined. At this point a loop is entered in which the population is subject to crossover, mutation, and fitness re-evaluation. This loop is repeated until one of the termination criteria is met.

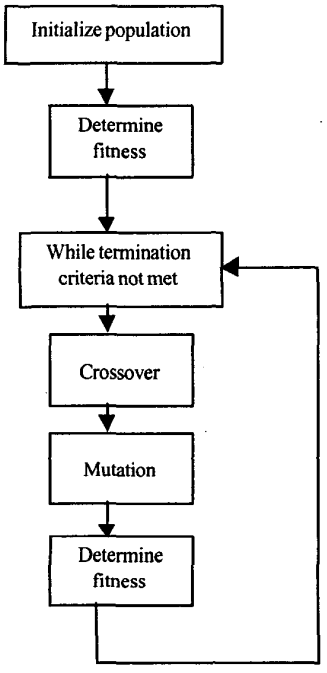

Figure 1: Flow chart for a typical genetic program

6.2 Initializing the population with a grow function

A population is constructed using two sets of objects, the terminal and the function sets, denoted as T and F, respectively. To make the initial population, we use a recursive function referred to as a grow function. First a random element of either set is selected to be the root. If the selected element is a terminal, the grow function stops. If it is a function, it produces random elements for arguments. Usually, the depth of the tree the function can grow is limited.

6.3 Crossover

In crossover a random node in each parent is selected to be the root of the subtree to exchange. The parental node is selected using a crossover probability. Parents can be selected through roulette wheel selection [29]. The entire subtree down from the crossover points are exchanged while ensuring the syntactic structure is preserved.

6.4 Mutation

In mutation a random node of the program is selected based on probability of mutation. That node and its subtree are deleted and replaced by a new function, created by the grow function in the same way the population is originally made. Even if the old subtree was simply a terminal, the new subtree can have multiple terminals and functions.

7 Building the fuzzy kinematic-ID subtree

This section discusses the automatic generation of a subtree of the RM's isolated platform decision tree (IPDT) using a GP to data mine a data base of military scenarios. The IPDT of the fuzzy RM has been extensively discussed [7,28]. The IPDT uses sensor data to make decisions about the threat status of an incoming platform. A platform may be an airplane, helicopter, ship, etc. Previous versions of the IPDT were constructed by hand based on human expertise.

7.1 Data base construction

As in any data mining operation the first step is the construction of the data base that will be mined. The data base used for automatic construction of subtrees of the IPDT consists of sensor output for the various platforms involved in the engagement. Each record contains the range, bearing, elevation, ID information for the emitting platforms involved, etc. It also contains a number between zero and one, which represents an expert's opinion as to whether or not the emitter is attacking.

7.2 Terminal and function sets

To use the genetic program it is necessary to construct terminal and function sets relevant to the problem. The terminal sets used for construction of subtrees of the IPDT typically consist of one or more fuzzy root concepts [7]. A typical terminal set might be

T={close, heading_in, ranging, banking, friend, lethal}. (1)

The root concept "close" is explained in reference [28]. The root concept of "heading_in" is similar to "close", but with range replaced by the emitter's heading in the root concept's membership function. "Ranging" and "banking" have fuzzy membership functions that are functions of the second time derivative of range and heading, respectively. The fuzzy membership function for "friend" gives the degree of membership of the detected platform in the concept "friend", i.e., how much confidence does blue have that the emitter is a friend. Finally, the membership function for the root concept "lethal" is found by summing the membership functions for all the foe classes. These concepts are explained in greater detail in the literature [28].

The function set, F, consist of the logical operations of "AND" and "OR" as well as the logical modifier "NOT", i.e.,

$$F=\{\text{AND, OR, NOT}\}. \quad (2)$$

7.3 The fitness function

The fitness function for data mining the IPDT subtree is

$$fitness(i) \equiv \frac{1}{n_{time} \cdot n_{db}} g(i, n_{db}, n_{time}, \tau) - \alpha \cdot l(i) \quad (3)$$

where

$$g(i, n_{db}, n_{time}, \tau) \equiv \quad (4)$$
$$\sum_{j=1}^{n_{db}} \sum_{k=1}^{n_{time}} \chi(\tau - | \mu_{gp}(i, t_k, e_j) - \mu_{expert}(t_k, e_j) |)$$

where e_j is the j^{th} element of the data base; t_k is the k^{th} time step; n_{db} is the number of elements in the data base; n_{time} is the number of time steps; τ is the tolerance; $\mu_{gp}(i,t_k,e_j)$ is the output of the fuzzy decision tree created by the GP for the i^{th} element of the population for time step t_k and data base element e_j; $\mu_{expert}(t_k,e_j)$ is an expert's estimate as to what the fuzzy decision tree should yield as output for time step t_k and data base element e_j; α is the parsimony coefficient; $l(i)$ is the length of the i^{th} element of the population, i.e., the number of nodes in the fuzzy decision tree corresponding to the i^{th} element; $\chi(t)$ is the Heaviside step function which is unity for $t \geq 0$ and zero otherwise.

Observe, that (4) reflects that the expert's estimate, $\mu_{expert}(t_k,e_j)$ is uncertain, and need only be reproduced within a tolerance, τ. Also, to increase the robustness of the GP created tree, the fitness of the fuzzy decision tree used by the GP is found by averaging over time and the database.

The parsimony pressure, $\alpha \bullet l(i)$, appearing on the right-hand-side of (3) provides a penalty that reduces the i^{th} population element's fitness if it is longer than needed. Thus given two trees that are both effective, the smaller tree will have the higher fitness. This provides a computational implementation of Occam's razor [30].

7.4 The termination criteria

The genetic program terminates after one of the following occurs: the number of generations reaches a preset maximum, the fitness has not changed by a preset amount in a certain number of generations or the fitness is within an acceptable tolerance of the maximum value.

7.5 Genetic program generated IPDT subtrees

Figure 2 depicts the IPDT subtree considered for construction using GP based data mining. This subtree was originally drawn based on experts' intuition. A line on a vertex denotes the logical connective "AND", a vertex without a line indicates the logical connective "OR", and a circle on an edge denotes the logical modifier "NOT". The tree is read as "if close and not a friend or heading_in and not a friend then attacking." The root concepts are "close", "heading_in" and "friend". The composite concept is "attacking".

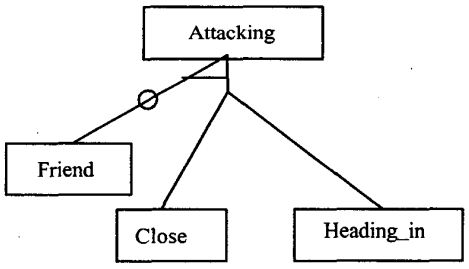

Figure 2: The IPDT subtree constructed by the GP

This subtree of the IPDT has been rediscovered by data mining a data base of military scenarios using a GP. Other more sophisticated trees have been discovered by GP based data mining, but this simple tree is considered here to illustrate the process. The fact that concepts like "ranging" and "banking" do not appear on the tree is related to the database used for the data mining procedure. There were no scenarios in the database that would make use of the "ranging" and "banking" concepts. The GP in many different runs was successful in constructing this subtree as expected, however, it did not always construct the same tree. Also, using different random seeds for each run, the number of generations required for the GP to stop varied. The GP's ability to construct the same subtree as

that written down based on experts' rules provides a form of support for the subtree since it can be found in multiple ways. Finally, the GP's ability to construct other trees points up the potential non-uniqueness of the subtree.

In one class of experiments the smallest number of generations required to generate the tree in Figure 2 was three and the maximum 30. The maximum number of generations the GP could have run was 100. The GP, in one case, generated a tree different from the anticipated result in Figure 2. This tree is under current examination and may prove to be superior to the subtree in Figure 2.

The GP's ability to find different fuzzy decision trees for the same problem most likely relates to the military data base that is being data mined, the fitness function and the parameters characterizing convergence. These are subjects of current research.

8 Summary

A fuzzy logic based resource manager (RM) for optimal allocation and scheduling of electronic attack resources distributed over many platforms is under development. Five components of the RM are discussed. Genetic algorithm based co-evolutionary data mining is examined. Co-evolution refers to a process where both friend and foe agents and meta-agents simultaneously evolve in a complex simulated environment perceived by various sensors. Construction of the database, which is used for data mining and optimization was summarized. Two methods of co-evolutionary optimization, computer versus computer (CVC) and human versus computer (HVC) optimization were discussed. CVC optimization involves evolution with a computer-controlled opponent(s); HVC optimization, with a human-controlled opponent or both human and computer controlled opponents. Experimental results for each form of co-evolutionary optimization were discussed and a comparison of both methods was outlined. It was found in HVC optimization that the human player quickly reached the limits of his or her expertise resulting in the RM's parameters reaching a constant value. In CVC optimization, the RM's computerized opponent proved more resilient than a human player resulting in blue and red parameters, which change rapidly in time, unlike in HVC mode. The more resilient computerized opponent in CVC mode exposed the RM to more types of strategies with the potential for a more adaptive and robust RM. Examples of the resource manager's multi-platform response are referenced to illustrate the RM's excellent performance and as a method of determining the value of the information data mined. A method for data mining fuzzy decision tree structure, and hence fuzzy if-then rules from military databases is introduced. This method uses a genetic program, an algorithm that automatically creates other computer programs, as a data mining function. The genetic program's structure is discussed as well as the terminal set, function set, the fitness function, termination criteria, population initialization, the operations of cross-over and mutation, and the construction of the data base used for data mining. The use of parsimony pressure to limit the length of the fuzzy decision tree while maintaining the tree's effectiveness is discussed. An explicit fitness function including parsimony pressure is examined. Finally, an example of a fuzzy decision tree generated by this algorithm is discussed.

9 Acknowledgements

This work was sponsored by the Office of Naval Research. The authors would also like to acknowledge Mr. Edward Khoury, Mr. Robert Rhyne, Ms. Kristin Fisher, Mr. Robert Xander, Dr. Joseph Lawrence III and Dr. Preston Grounds.

References

[1] D. C. Schleher, *Electronic Warfare in the Information Age*, Chapter 1, Artech House, Boston, 1999.

[2] J.M. Molina Lopez, F.J. Jimenez Rodriguez, and J.R. Casar Corredera, "Symbolic Processing for Coordinated Task Management in Multiradar Surveillance Networks," *Fusion98, Proceedings of the International Conference on Multisource-Multisensor Information Fusion*, R. Hamid, and D. Zhu, Vol . II, pp. 725-732, CSREA Press, Las Vegas, 1998.

[3] L.H. Tsoukalas and R.E. Uhrig, *Fuzzy and Neural Approaches in Engineering*, Chapter 5, John Wiley and Sons, New York, 1997.

[4] J.P. Bigus, *Data Mining with Neural Nets*, Chapter 1, McGraw-Hill, New York, 1996.

[5] J. H. Holland, *Hidden Order How Adaptation Builds Complexity* , pp. 1-15, Perseus Books, Reading, 1995.

[6] H. J. Zimmerman, *Fuzzy Set Theory and its Applications*, p. 11, Kluwer Academic Publishers Group, Boston, 1991.

[7] J.F. Smith, III and R. Rhyne, II, "A Resource Manager for Distributed Resources: Fuzzy Decision Trees and Genetic Optimization," *Proceeding of the International Conference on Artificial Intelligence, IC-AI'99,* H. Arabnia, Vol. II, pp. 669-675, CSREA Press, Las Vegas, 1999.

[8] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning,* Addison-Wesley, Reading, 1989.

[9] J.C. Bezdek and J.C.Dunn, "Optimal Fuzzy Partitions: A Heuristic for Estimating the Parameters in a Mixture of Normal Distributions", IEEE Trans. Comp., **Vol C-24**, pp. 835-838, 1975.

[10] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.

[11] J. Smith, A. Huynh, M. Kim, "An Application of Clustering and Speed Discrimination to Tracking,"NRL Formal Report, NRL/FR/5740--95-9801, Naval Research Laboratory, Washington D.C. 20375-5000, December 1995.

[12] J. F. Smith III, "A Fuzzy Clustering and Superclustering Scheme for Extracting Structure from Data," NRL Formal Report, NRL/FR/5740--96-9844, Naval Research Laboratory, Washington D.C. 20375-5000, December 1996.

[13] G.V. Trunk and J.D. Wilson, "Association of DF Bearing Measurements with Radar Track," IEEE Trans on AES, **Vol. AES-23., No. 4**, pp 438-447, 1987.

[14] G.V. Trunk and J.O. Coleman, "Radar-ESM Correction Decision Procedure," NRL Report 8476, Naval Research Laboratory, Washington D.C. 20375-5000, May 7, 1981.

[15] J. F. Smith III, "A Fuzzy Logic Multisensor Association Algorithm," *Signal Processing, Sensor Fusion, and Target Recognition VI*, I. Kadar, Vol. 3068, pp. 76-87, SPIE Proceedings, Orlando, 1997.

[16] J. F. Smith III, "A Fuzzy Logic Multisensor Association Algorithm: Theory and Simulation," NRL Formal Report, NRL/FR/5740--97-9866, Naval Research Laboratory, Washington D.C. 20375-5000, September 1997.

[17] J. F. Smith III, "A Fuzzy Logic Multisensor Association Algorithm: Dealing with Multiple Targets, Intermittent Data and Noise," *Signal Processing, Sensor Fusion, and Target Recognition VI*, I. Kadar, Vol. 3374, pp. 2-13, SPIE Proceedings, Orlando, 1998.

[18] J. F. Smith III, "Fuzzy Logic Association: Performance, Implementation Issues, and Automated Resource Allocation", *Signal Processing, Sensor Fusion, and Target Recognition VIII*, I. Kadar, Vol. 3720, pp. 228-238, SPIE Proceedings, Orlando, 1999.

[19] J.F. Smith III, "A Fuzzy Logic Multisensor Association Algorithm: Applied to Noisy, Intermittent and Sparse Data", *Fusion98, Proceedings of the International Conference on Multisource-Multisensor Information Fusion*, R. Hamid and D. Zhu, pp. 681-688, CSREA Press, Las Vegas,1998.

[20] J.F. Smith III," A Fuzzy Logic Multisensor Association Algorithm: Multiple Emitters, Computational Complexity, and Noisy Data", NRL Formal Report NRL/FR/5740--99-9918, Naval Research Laboratory, Washington D.C. 20375-5000, July 15, 1999.

[21] D. Cliff and G. F. Miller, "Co-evolution of Pursuit and Evasion II: Simulation Methods and Results," *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB96)*, P. Maes, M. Mataric, J.-A. Meyer, J. Pollack, and S.W. Wilson (eds.),. pp. 1-10, MIT Press Bradford Books, Cambridge, 1996.

[22] James F. Smith III and Robert D. Rhyne II, "A Fuzzy Logic Algorithm for Optimal Allocation of Distributed Resources", *Fusion 99: Proceednings of the Second International Conference on Information Fusion*, D. Zhu, pp 402-409, International Society of Information Fusion, San Jose, 1999.

[23] J. F. Smith III and R.D. Rhyne II, "Fuzzy logic based resource allocation for isolated and multiple platforms", *Signal Processing, Sensor Fusion, and Target Recognition IX*, I. Kadar, Vol. 4052, pp. 36-47, SPIE Proceedings, Orlando, 2000.

[24] J. F. Smith III and R.D. Rhyne II, "Genetic algorithm based optimization of a fuzzy logic resource manager for electronic attack", *Data Mining and Knowledge Discovery II*, B.Dasarathy, Vol. 4057, pp. 62-73, SPIE Proceedings, Orlando, 2000.

[25] J. F. Smith III and R. D. Rhyne II, "A Fuzzy Logic Resource Manager and Underlying Data Mining Techniques", *Fusion2000: Proceedings of the 3rd International Conference on Information Fusion*, R. Reynaud, Vol. II, pp. WEB1-3 – WEB1-9, International Society of Information Fusion, Paris, 2000.

[26] J.F. Smith, III and R.D. Rhyne, II, "Genetic Algorithm Based Optimization of a Fuzzy Logic Resource Manager: Data Mining and Co-evolution," *Proceeding of the International Conference on Artificial Intelligence, IC-AI'2000*, H. Arabnia, Vol. I, pp 421-428, CSREA Press, Las Vegas, 2000.

[27] J.F. Smith III and R. Rhyne II,"Optimal Allocation of Distributed Resources Using Fuzzy Logic and a Genetic Algorithm", NRL Formal Report NRL/FR/5741-00-9970, Naval Research Laboratory, Washington D.C. 20375-5000, September 29, 2000.

[28] J.F. Smith, III and R. D. Rhyne II, "Fuzzy logic resource manager: tree structure and optimization", *Signal Processing, Sensor Fusion, and Target Recognition X*, I. Kadar, Vol. 4380, pp.312-323, SPIE Proceedings, Orlando, 2001.

[29] J.R., Koza, , F.H. Bennett III, D. Andre, and M.A. Keane. *Genetic Programming III: Darwinian Invention and Problem Solving*. Chapter 2, Morgan Kaufmann Publishers, San Francisco, 1999.

[30] J. Gribbin, *Companion to the Cosmos*, p. 299, The Orion Publishing Group Ltd, London, 1996.